

Computation of Pseudospectral Abscissa for Large Scale Nonlinear Eigenvalue Problems

Karl Meerbergen[†] Emre Mengi[‡] Wim Michiels[†]
Roel Van Beeumen[†]

August 18, 2015

Abstract

We present an algorithm to compute the pseudospectral abscissa for a nonlinear eigenvalue problem. The algorithm relies on global under-estimator and over-estimator functions for the eigenvalue and singular value functions involved. These global models follow from eigenvalue perturbation theory. The algorithm has three particular features. First, it converges to the globally rightmost point of the pseudospectrum, and it is immune to nonsmoothness. Second, it is applicable to large scale problems since the dominant cost per iteration stems from computing the smallest singular value and associated singular vectors, for which efficient iterative solvers can be used. Furthermore, a significant increase in computational efficiency can be obtained by subspace acceleration, i.e., by restricting the domains of the linear maps associated with the matrices involved to small but suitable subspaces, and solving the resulting reduced problems. Occasional restarts of these subspaces further enhance the efficiency for large scale problems. Finally, in contrast to existing iterative approaches based on constructing low rank perturbations and rightmost eigenvalue computations, the algorithm only relies on computing singular values of complex matrices. Hence, the algorithm does not require solutions of nonlinear eigenvalue problems, thereby further increasing efficiency and reliability. This work is accompanied by a robust implementation of the algorithm, that is publicly available.

Keywords: pseudospectra, nonlinear eigenvalue problem, eigenvalue perturbation theory, nonsmooth optimization, subspace methods, global optimization

AMS subject classifications: 65F15, 90C30, 65H20

1 Introduction

We consider the nonlinear eigenvalue problems of the form

$$F(\lambda)x = 0, \tag{1.1}$$

where $F : \Omega \rightarrow \mathbb{C}^{n \times n}$ is an analytic matrix-valued function on $\Omega \subseteq \mathbb{C}$. The scalar $\lambda \in \Omega$ satisfying the equation above for an $x \in \mathbb{C}^n \setminus \{0\}$ is called an eigenvalue, while x is called the corresponding eigenvector. Such eigenvalue problems when F is a matrix polynomial, especially the quadratic eigenvalue problem, arise from various engineering applications for instance from applications in structural design and fluid mechanics [24]. Nonpolynomial nonlinear eigenvalue problems are also of great interest: finite-element discretizations of boundary value problems,

[†]Department of Computer Science, KU Leuven, University of Leuven, 3001 Heverlee, Belgium (Roel.VanBeeumen@cs.kuleuven.be, Karl.Meerbergen@cs.kuleuven.be, Wim.Michiels@cs.kuleuven.be). The work of the authors were supported in part by OPTEC, the Optimization in Engineering Center of KU Leuven.

[‡]Department of Mathematics, Koç University, Rumelifeneri Yolu, 34450 Sarıyer-İstanbul, Turkey (emengi@ku.edu.tr). The work of the author was supported in part by the European Commission grant PIRG-GA-268355, TUBITAK - FWO (Scientific and Technological Research Council of Turkey - Belgian Research Foundation, Flanders) joint grant 113T053, and by the BAGEP program of Turkish Academy of Science.

for instance in photonics, lead to eigenvalue problems of the form (1.1), where $F(\lambda)$ is a rational function of λ [17, 5]; delay systems in control theory necessitate nonlinear eigenvalue problems where $F(\lambda)$ involves exponentials of λ [22]. For recent progresses on the topic, we refer to the survey paper [17] and the thesis works [5, 26].

Stability of the continuous dynamical system associated with the nonlinear eigenvalue problem is a fundamental issue. In terms of the nonlinear eigenvalue problem (1.1), this amounts to the inclusion of all of the eigenvalues on the left half of the complex plane. However, the system is often subject to uncertainties, thus it is often desired that a system remains stable under small perturbations of parameters. This is reflected into the nonlinear eigenvalue problem (1.1) as the inclusion of the eigenvalues of the original problem as well as all perturbed problems in the left half of the complex plane. Moreover, a stable system can still exhibit transient behavior before reaching the equilibrium eventually. For instance, for the standard eigenvalue problem $Av = \lambda v$ and the associated dynamical system $x'(t) = Ax(t)$, this is explained by the Kreiss matrix theorem [25, Theorem 18.5]. A corollary of this theorem is that a stable system $x'(t) = Ax(t)$ becoming unstable under small perturbations of A must exhibit transient growth.

For robustness against uncertainties and to assess the transient behavior of solutions of a stable system, a modern approach is the consideration of the ϵ -pseudospectrum of F . This is the set in the complex plane to which the eigenvalues of F can be shifted when perturbations at a distance ϵ or closer are taken into account. The ϵ -pseudospectral abscissa, the supremum of the real parts of the elements of the ϵ -pseudospectrum, constitutes a uniform bound on the asymptotic growth rate of the solutions for all perturbations at a distance ϵ or closer. Consequently, it assesses robust stability [2]. The ϵ -pseudospectral abscissa is also closely related to the distance to instability [27] and the H-infinity norm of transfer functions defined appropriately (see [30] for relations between H-infinity norms and robust stability criteria).

The types of nonlinear eigenvalue problems we consider can often be reformulated as linear eigenvalue problems (for example by a so-called linearization for a polynomial eigenvalue problem, or by a transformation to an equivalent infinite dimensional operator eigenvalue problem for the delay eigenvalue problem [10]). But we will not consider the unstructured pseudospectra of a particular type of linearization. Instead, as in [23, 21, 27], we will explicitly take the structure of the original nonlinear eigenvalue problem into account in the definition of its pseudospectra.

1.1 Formal Definition

Formally, the analytic matrix-valued function F can always be expressed of the form

$$F(\lambda) = \sum_{j=0}^m f_j(\lambda) A_j \quad (1.2)$$

where $A_j \in \mathbb{C}^{n \times n}$, the scalar function $f_j : \Omega \rightarrow \mathbb{C}$ is analytic on its entire domain Ω for $j = 0, \dots, m$, and $m \leq n^2 - 1$. The spectrum of F given by

$$\Lambda(F) := \left\{ \lambda \in \mathbb{C} : \det \left(\sum_{j=0}^m f_j(\lambda) A_j \right) = 0 \right\}, \quad (1.3)$$

more specifically the *spectral abscissa*

$$\alpha(F) := \sup \{ \Re \lambda : \lambda \in \Lambda(F) \} \quad (1.4)$$

is responsible for the asymptotic behavior of the associated dynamical system, yet it does not say much about the transient behavior by itself.

To take the uncertainties and transient behavior into account, we are interested in the perturbed eigenvalue problem

$$\left(\sum_{j=0}^m f_j(\lambda) (A_j + \delta A_j) \right) x = 0. \quad (1.5)$$

We quantify the distance between the original matrix-valued function (1.2), and the perturbed one in (1.5) by introducing the norm

$$\|\Delta\|_{\text{glob}} := \left\| \begin{bmatrix} w_0 \|\delta A_0\|_2 \\ \vdots \\ w_m \|\delta A_m\|_2 \end{bmatrix} \right\|_{\infty}, \quad (1.6)$$

where $\Delta := (\delta A_0, \dots, \delta A_m) \in \mathbb{C}^{n \times n \times (m+1)}$ for given non-negative real scalars w_j (possibly ∞) for $j = 0, \dots, m$, equivalently we equip the vector space of analytic matrix-valued functions of the form (1.2) with variable coefficient matrices A_j but fixed scalar functions f_j with a norm.

We then define the ϵ -pseudospectrum of F by

$$\Lambda_{\epsilon}(F) := \bigcup_{\|\Delta\|_{\text{glob}} \leq \epsilon} \left\{ \lambda \in \mathbb{C} : \det \left(\sum_{j=0}^m f_j(\lambda) (A_j + \delta A_j) \right) = 0 \right\}, \quad (1.7)$$

and the ϵ -pseudospectral abscissa by

$$\alpha_{\epsilon}(F) := \sup \{ \Re \lambda : \lambda \in \Lambda_{\epsilon}(F) \} \quad (1.8)$$

as an indicator of the transient behavior and a measure of robust stability of the dynamical system associated with (1.1). The following characterization of $\Lambda_{\epsilon}(F)$ was derived in [20].

Proposition 1.1.

$$\Lambda_{\epsilon}(F) = \left\{ \lambda \in \mathbb{C} : \sigma_{\min} \left(\sum_{j=0}^m f_j(\lambda) A_j \right) \leq \epsilon \|w(\lambda)\|_1 \right\}, \quad (1.9)$$

where $\sigma_{\min}(\cdot)$ denotes the smallest singular value of its matrix argument, and

$$w(\lambda) := \begin{bmatrix} \frac{f_0(\lambda)}{w_0} & \dots & \frac{f_m(\lambda)}{w_m} \end{bmatrix}^T. \quad (1.10)$$

Throughout the text, we assume that the portion of $\Lambda_{\epsilon}(F)$ to the right-hand side of each vertical line in the complex plane is bounded. Formally, letting

$$\mathbb{C}_{\geq \delta} := \{z \in \mathbb{C} : \Re z \geq \delta\}$$

for a given $\delta \in \mathbb{R}$, it is assumed that $\Lambda_{\epsilon}(F) \cap \mathbb{C}_{\geq \delta}$ is bounded for all $\delta \in \mathbb{R}$. This assumption ensures the well-posedness of $\alpha_{\epsilon}(F)$ defined by (1.8). For a thorough discussion on this condition, we refer to [21].

1.2 Literature

The ϵ -pseudospectrum for matrices, that is when $F(\lambda) = \lambda I - A$, has been popularized by Trefethen in the last two decades [25]. In the nonlinear case, earlier works focused on the pseudospectra of matrix polynomials [23, 9, 13]. More recent research concentrated on the nonpolynomial setting, either the delay system setting [6] or the more general setting [20, 28].

Particular attention has been paid to the computation of the pseudospectral abscissa. For the pseudospectral abscissa of a matrix, the first globally convergent algorithm was proposed in [2]. Since every iteration of this algorithm requires computing all eigenvalues of a matrix of twice the dimensions of the original matrix, it is restricted to problems of moderate size. In [7], a locally convergent algorithm for large scale matrices is proposed, where every iteration relies on computing the rightmost eigenvalue of the original matrix plus a rank one perturbation (see also [11] for an improvement of this algorithm based on subspace acceleration). The algorithm of [7] has been extended to nonlinear eigenvalue problems in [21], and it has also been adopted to compute the distance to instability from a nonlinear dynamical system in [27]. In the next subsection we situate the proposed algorithm with respect to these works.

1.3 Motivation and Outline

We present a globally convergent algorithm for the computation of $\alpha_\epsilon(F)$, particularly suitable for large scale problems, that is when A_j are large matrices. Three main components of the algorithm are introduced in [18, 19, 11]. In [18], a locally convergent algorithm is presented for optimizing a linear function subject to a constraint on a smallest eigenvalue function. The algorithm is immune to the nonsmooth nature of the smallest eigenvalue function. We describe how this algorithm can be adopted to compute $\alpha_\epsilon(F)$ based on the characterization (1.9) of $\Lambda_\epsilon(F)$ in Section 2.

Unfortunately, this yields a locally rightmost point, which is possibly not rightmost globally. We overcome this by performing a vertical search by means of the algorithm introduced in [19] for the global optimization of a prescribed eigenvalue of a Hermitian and analytic matrix-valued function. We fix the real part α of the locally rightmost point, and perform the minimization of

$$\frac{\sigma_{\min} \left(\sum_{j=0}^m f_j(\alpha + i\omega) A_j \right)}{\|w(\alpha + i\omega)\|_1} \quad (1.11)$$

over all $\omega \in \mathbb{R}$ globally. If this globally minimal value is less than ϵ , then we repeat the local search starting from $\alpha + i\omega_*$ where ω_* is a global minimizer of (1.11). We refer to Figure 5 (in the numerical examples section towards the end of this text) for an illustration of the interplay between the local searches and vertical searches. In this illustration, local searches yield locally (but not globally) rightmost points twice. In each of these two cases, a vertical search provides a point strictly inside $\Lambda_\epsilon(F)$ whose real part is the same as the locally rightmost point. The vertical search idea is discussed in Section 3.

Due to the fact that the computational cost is dominated by computing the smallest singular value and corresponding singular vectors, for which fast iterative methods are amendable, the proposed algorithm is applicable to large scale problems. Moreover, a significant speed-up can be achieved by incorporating a subspace restriction, whose idea is originally proposed in [11] for the computation of the pseudospectral abscissa of a matrix. The remarkable low rank property observed and exploited in that paper still holds in this more general nonlinear eigenvalue setting. In particular, there exists a one dimensional subspace of \mathbb{C}^n such that the ϵ -pseudospectral abscissa of $F(\lambda)$ remains the same when the domain of the map $v \mapsto F(\lambda)v$ is restricted to this one dimensional subspace. The details of this subspace idea for nonlinear eigenvalue problems are worked out in Section 4. The overall idea is to restrict the domain of $v \mapsto F(\lambda)v$ to very low dimensional subspaces of \mathbb{C}^n , and compute the ϵ -pseudospectral abscissa of the resulting smaller problems by means of the locally convergent algorithm in [18]. The vertical searches are performed on the original $F(\lambda)$. This is justified by the rare need for these vertical searches.

One genuine aspect of the algorithm is an occasional restart strategy for the subspaces as argued in Section 5. Since the essential task is to determine or capture a one dimensional subspace, the algorithm erases off the old subspaces occasionally. Thus after a vertical search, if a further application of the local algorithm is deemed to be necessary, the algorithm starts with a one dimensional subspace from scratch. Moreover, when the dimension of the subspace becomes large enough (still considerably smaller than n), the algorithm keeps only the lastly added one dimensional subspace discarding the rest.

The overall framework is outlined in Algorithm 4. This algorithm features favorable properties over existing algorithms, for instance [21]. Specifically, **(i)** it converges globally rather than locally; **(ii)** it is immune to nonsmoothness, that is, even if $\alpha_\epsilon(F)$ is attained at a point say $z_* \in \mathbb{C}$ where $\sigma_{\min}(F(z_*))$ is not simple, it still converges; **(iii)** it handles large scale problems well. The subspace method coupled with the restart strategy contributes to this largely, but the restarts would not be as effective without vertical searches that are performed globally; **(iv)** in contrast to the approach of [21], the algorithm does not rely on a nonlinear eigenvalue solver (provided it is initialized with the rightmost eigenvalue): instead of the rightmost eigenvalue of perturbed nonlinear eigenvalue problems, it is based on the repeated computation of the smallest singular value of complex matrices.

2 Determination of Locally Rightmost Points

Due to Proposition 1.9, the ϵ -pseudospectral abscissa of F can be cast as the following constrained eigenvalue optimization problem:

$$\begin{aligned} & \text{maximize } \Re z \\ & z \in \mathbb{C} \\ & \text{subject to } \lambda(\Re z, \Im z) := \lambda_{\min}[F(\Re z, \Im z)^* F(\Re z, \Im z)] - \epsilon^2 \|w(\Re z, \Im z)\|_1^2 \leq 0, \end{aligned} \quad (2.1)$$

where we view the matrix-valued function in (1.2) as $F : \mathbb{R}^2 \rightarrow \mathbb{C}^{n \times n}$ and the weight function in (1.10) as $w : \mathbb{R}^2 \rightarrow \mathbb{R}$, by associating \mathbb{R}^2 with \mathbb{C} . Throughout the text, to ease the notation $F(\cdot)$, $\lambda(\cdot)$, $w(\cdot)$ and $f_j(\cdot)$, $j = 0, \dots, m$ represent both the functions from \mathbb{C} and the functions from \mathbb{R}^2 . Whether the function with domain \mathbb{C} or the function with domain \mathbb{R}^2 is referred to in a specific instance will be clear from the context. Furthermore, in (2.1) and in what follows the notations $\lambda_{\min}[\cdot]$ and $\lambda_{\max}[\cdot]$ represent the smallest eigenvalue and the largest eigenvalue of the matrix argument, respectively.

An approach to maximize a linear objective subject to a smallest eigenvalue constraint was suggested in [18]. Below, we describe how this approach can be extended to deal with (2.1), in particular the additional nonsmoothness due to $\|w(\Re z, \Im z)\|_1^2$, which occurs whenever $f_j(\Re z, \Im z) = 0$ for some $j \in \{0, \dots, m\}$. The extension relies on the global over-estimators for $\lambda(\cdot)$ of the form specified in Theorem 2.1 below.

Theorem 2.1. *Suppose $z_k = (z_{k1}, z_{k2}) \in \mathbb{R}^2$ is a point such that $\sigma_{\min}[F(z_k)]$ is simple, and $f_j(z_k) \neq 0$ for each j . Let $\gamma := \gamma_\lambda + \epsilon^2 \gamma_w$, where γ_λ is a scalar satisfying*

$$\lambda_{\max} \{ \nabla^2 \lambda_{\min}[F(\Re z, \Im z)^* F(\Re z, \Im z)] \} \leq \gamma_\lambda \quad (2.2)$$

for all $z \in \mathbb{C}$ where $\sigma_{\min}[F(\Re z, \Im z)]$ is simple, and γ_w is a scalar satisfying

$$\| \nabla^2 [\|w(\Re z, \Im z)\|_1^2] \|_2 \leq \gamma_w \quad (2.3)$$

for all $z \in \mathbb{C}$ where $f_j(\Re z, \Im z) \neq 0$ for each j . We have

$$\lambda(\Re z, \Im z) \leq q_k(\Re z, \Im z) := \lambda_k + \nabla \lambda_k^T ((\Re z, \Im z) - z_k) + \frac{\gamma}{2} \|(\Re z, \Im z) - z_k\|_2^2 \quad \forall z \in \mathbb{C}$$

where $\lambda_k := \lambda(z_{k1}, z_{k2})$ and $\nabla \lambda_k := \nabla \lambda(z_{k1}, z_{k2})$.

For the proof of Theorem 2.1, we refer to the proof of Theorem 2.2 in [18] noting that

$$\begin{aligned} \lambda_{\max} [\nabla^2 \lambda(\Re z, \Im z)] & \leq \lambda_{\max} \{ \nabla^2 \lambda_{\min}[F(\Re z, \Im z)^* F(\Re z, \Im z)] \} + \lambda_{\max} \{ \nabla^2 [-\epsilon^2 \|w(\Re z, \Im z)\|_1^2] \} \\ & \leq \lambda_{\max} \{ \nabla^2 \lambda_{\min}[F(\Re z, \Im z)^* F(\Re z, \Im z)] \} + \epsilon^2 \| \nabla^2 [\|w(\Re z, \Im z)\|_1^2] \|_2 \\ & \leq \gamma_\lambda + \epsilon^2 \gamma_w = \gamma \end{aligned}$$

for each $z \in \mathbb{C}$. Replacing the eigenvalue constraint in (2.1) with the over-estimator of Theorem 2.1 results in the following convex and smooth problem:

$$\begin{aligned} & \text{maximize } \Re z \\ & z \in \mathbb{C} \\ & \text{subject to } \lambda_k + \nabla \lambda_k^T ((\Re z, \Im z) - z_k) + \frac{\gamma}{2} \|(\Re z, \Im z) - z_k\|_2^2 \leq 0. \end{aligned} \quad (2.4)$$

The algorithm generates a sequence $\{z_k\}$ in \mathbb{R}^2 such that z_{k+1} is the maximizer of (2.4) given z_k . Since the feasible set of (2.4) (a disk) is a subset of the feasible set of the original problem (2.1), each z_k remains feasible with respect to the original problem provided z_0 is feasible. By applying the first order optimality conditions to (2.4), two consecutive iterates in the sequence $\{z_k\}$ are tied by the recurrence

$$z_{k+1} = z_k + \frac{1}{\gamma} \left[\frac{1}{\mu_+} \cdot (1, 0) - \nabla \lambda_k \right], \quad \text{where } \mu_+ = \frac{1}{\sqrt{\|\nabla \lambda_k\|_2^2 - 2\gamma \lambda_k}}, \quad (2.5)$$

$$\nabla \lambda_k = \begin{bmatrix} \text{Real} \left(v_k^* \frac{\partial F(z_k)^*}{\partial \Re z} F(z_k) v_k + v_k^* F(z_k)^* \frac{\partial F(z_k)}{\partial \Re z} v_k \right) - 2\epsilon^2 \|w(z_k)\|_1 \frac{\partial \|w(z_k)\|_1}{\partial \Re z} \\ \text{Real} \left(v_k^* \frac{\partial F(z_k)^*}{\partial \Im z} F(z_k) v_k + v_k^* F(z_k)^* \frac{\partial F(z_k)}{\partial \Im z} v_k \right) - 2\epsilon^2 \|w(z_k)\|_1 \frac{\partial \|w(z_k)\|_1}{\partial \Im z} \end{bmatrix},$$

$$\frac{\partial \|w(z_k)\|_1}{\partial \Re z} = \sum_{j=0}^m \frac{1}{w_j} \frac{\partial |f_j(z_k)|}{\partial \Re z}, \quad \frac{\partial \|w(z_k)\|_1}{\partial \Im z} = \sum_{j=0}^m \frac{1}{w_j} \frac{\partial |f_j(z_k)|}{\partial \Im z},$$

and $v_k \in \mathbb{C}^n$ is a unit right singular vector corresponding to $\sigma_{\min}[F(z_k)]$. Here, we benefit from the analytical formulas for the derivatives of eigenvalue functions [12], in particular to calculate the derivatives of $\lambda_{\min}[F(\Re z, \Im z)^* F(\Re z, \Im z)]$. Recurrence (2.5) holds under the assumption that $\nabla q_k(z_{k+1}) \neq 0$. The condition $\nabla q_k(z_{k+1}) = 0$ is rather unlikely, it occurs only if $\nabla \lambda_k = 0$ and $\lambda_k = 0$ (see Theorem 2.3 in [18]).

2.1 Upper Bounds on Second Derivatives

In this section, we present bounds γ_λ and γ_w satisfying (2.2) and (2.3), respectively. An application of Theorem 6.1 in [18] yields

$$\lambda_{\max} \{ \nabla^2 \lambda_{\min} [F(\Re z, \Im z)^* F(\Re z, \Im z)] \} \leq \lambda_{\max} \{ \nabla^2 [F(\Re z, \Im z)^* F(\Re z, \Im z)] \} \quad (2.6)$$

for $z \in \mathbb{C}$ such that $\sigma_{\min}[F(z)]$ is simple, where

$$\nabla^2 [F(\Re z, \Im z)^* F(\Re z, \Im z)] := \begin{bmatrix} \frac{\partial^2 [F(\Re z, \Im z)^* F(\Re z, \Im z)]}{\partial \Re z^2} & \frac{\partial^2 [F(\Re z, \Im z)^* F(\Re z, \Im z)]}{\partial \Re z \partial \Im z} \\ \frac{\partial^2 [F(\Re z, \Im z)^* F(\Re z, \Im z)]}{\partial \Im z \partial \Re z} & \frac{\partial^2 [F(\Re z, \Im z)^* F(\Re z, \Im z)]}{\partial \Im z^2} \end{bmatrix}.$$

For the standard ϵ -pseudospectral abscissa of a matrix A , i.e., when $F(z) = A - zI$, we have $\nabla^2 [F(\Re z, \Im z)^* F(\Re z, \Im z)] = 2I$. Consequently, inequality (2.6) leads to the upper bound $\lambda_{\max} \{ \nabla^2 \lambda_{\min} [F(\Re z, \Im z)^* F(\Re z, \Im z)] \} \leq 2$ for all $z \in \mathbb{C}$ such that $\sigma_{\min}[F(z)]$ is simple. In the general nonlinear setting (1.2), routine calculations yield

$$\nabla^2 [F(\Re z, \Im z)^* F(\Re z, \Im z)] = \sum_{k=0}^m \sum_{j=0}^m \left[\overline{F_{k,j}(\Re z, \Im z)} + F_{j,k}(\Re z, \Im z) \right] \otimes A_k^* A_j,$$

where

$$\begin{aligned} F_{j,k}(\Re z, \Im z) &:= \nabla^2 \overline{f_k(\Re z, \Im z)} \cdot f_j(\Re z, \Im z) + \nabla \overline{f_k(\Re z, \Im z)} \cdot \nabla f_j(\Re z, \Im z)^T \\ &= \overline{f_k''(z)} f_j(z) \begin{bmatrix} 1 & -i \\ -i & -1 \end{bmatrix} + \overline{f_k'(z)} f_j'(z) \begin{bmatrix} 1 & i \\ -i & 1 \end{bmatrix}, \end{aligned} \quad (2.7)$$

and \otimes denotes the Kronecker product. From here, by employing (2.6) and exploiting

$$\overline{F_{k,j}(\Re z, \Im z)} + F_{j,k}(\Re z, \Im z) = \left[\overline{F_{j,k}(\Re z, \Im z)} + F_{k,j}(\Re z, \Im z) \right]^*,$$

we deduce the following bound.

Theorem 2.2. *Suppose that $z \in \mathbb{C}$ is such that $\sigma_{\min}[F(z)]$ is simple. The following holds:*

$$\lambda_{\max} \{ \nabla^2 \lambda_{\min} [F(\Re z, \Im z)^* F(\Re z, \Im z)] \} \leq 2 \cdot \sum_{k=0}^m \sum_{j=0}^m \|F_{k,j}(\Re z, \Im z)\|_2 \cdot \|A_k^* A_j\|_2.$$

Example (Polynomial Eigenvalue Problem): Consider $F(z) = \sum_{j=0}^m z^j A_j$ for given matrices $A_j \in \mathbb{C}^{n \times n}$ for $j = 0, \dots, m$. Noting that $f_j(z) = z^j$, Theorem 2.2 combined with expression (2.7) for $F_{j,k}(\Re z, \Im z)$ would imply

$$\begin{aligned} \lambda_{\max} \{ \nabla^2 \lambda_{\min} [F(\Re z, \Im z)^* F(\Re z, \Im z)] \} &\leq \\ &4 \cdot \left[\sum_{k=0}^m \sum_{j=2}^m j \cdot (j-1) \cdot |z|^{j+k-2} \|A_k^* A_j\|_2 + \sum_{k=1}^m \sum_{j=1}^m k \cdot j \cdot |z|^{j+k-2} \|A_k^* A_j\|_2 \right]. \end{aligned}$$

Assuming that the ϵ -pseudospectrum of F is bounded and contained inside a ball of radius δ in the complex plane, we could set

$$\gamma_\lambda := 4 \cdot \left[\sum_{k=0}^m \sum_{j=2}^m j \cdot (j-1) \cdot \delta^{j+k-2} \|A_k^* A_j\|_2 + \sum_{k=1}^m \sum_{j=1}^m k \cdot j \cdot \delta^{j+k-2} \|A_k^* A_j\|_2 \right].$$

Similarly, bounds could be also derived for delay and rational eigenvalue problems based on Theorem 2.2.

The following bound is the consequence of rudimentary calculations.

Theorem 2.3. *Suppose $z \in \mathbb{C}$ is such that $f_j(z) \neq 0$ for $j = 0, \dots, m$. The following holds:*

$$\|\nabla^2 [\|w(\Re z, \Im z)\|_1^2]\|_2 \leq 2 \cdot \left[\sum_{j=0}^m \frac{1}{w_j} |f_j'(z)| \right]^2 + 2 \cdot \left[\sum_{j=0}^m \frac{1}{w_j} |f_j(z)| \right] \cdot \left[\sum_{j=0}^m \frac{1}{w_j} \left\{ 3 \cdot \frac{|f_j'(z)|^2}{|f_j(z)|} + |f_j''(z)| \right\} \right]$$

For instance, for the matrix polynomial $F(z) = \sum_{j=0}^m z^j A_j$ with $f_j(z) = z^j$, Theorem 2.3 gives rise to the bound

$$\|\nabla^2 [\|w(\Re z, \Im z)\|_1^2]\|_2 \leq 2 \cdot \left[\sum_{j=1}^m \frac{j \cdot |z|^{j-1}}{w_j} \right]^2 + 2 \cdot \left[\sum_{j=0}^m \frac{|z|^j}{w_j} \right] \cdot \left[\sum_{j=1}^m \frac{3j^2 \cdot |z|^{j-2}}{w_j} + \sum_{j=2}^m \frac{j \cdot (j-1) \cdot |z|^{j-2}}{w_j} \right].$$

If the ϵ -pseudospectral abscissa of F is contained inside the ball of radius δ , we could choose

$$\gamma_w := 2 \cdot \left[\sum_{j=1}^m \frac{j \cdot \delta^{j-1}}{w_j} \right]^2 + 2 \cdot \left[\sum_{j=0}^m \frac{\delta^j}{w_j} \right] \cdot \left[\sum_{j=1}^m \frac{3j^2 \cdot \delta^{j-2}}{w_j} + \sum_{j=2}^m \frac{j \cdot (j-1) \cdot \delta^{j-2}}{w_j} \right].$$

2.2 Convergence

Let us denote the components of $z_k \in \mathbb{R}^2$ with z_{k1} and z_{k2} . We call $\mathcal{C}(z_k) := z_{k1} + iz_{k2}$ the *complexification* of z_k . The sequence $\{z_{k1}\}$ is monotone increasing. This is because z_{k+1} is chosen among all $(\Re z, \Im z)$ satisfying $q_k(\Re z, \Im z) \leq 0$ such that $\Re z$ is as large as possible, and in particular z_k satisfies $q_k(z_k) = 0$. Additionally, since it is assumed that $\Lambda_\epsilon(F) \cap \mathbb{C}_{\geq \delta}$ is bounded for all $\delta \in \mathbb{R}$, the sequence $\{z_{k1}\}$ is bounded above. This would imply the convergence of $\{z_{k1}\}$ as stated next.

Theorem 2.4. *Suppose that $\Lambda_\epsilon(F) \cap \mathbb{C}_{\geq \delta}$ is bounded for all $\delta \in \mathbb{R}$, $\sigma_{\min}[F(z_k)]$ is simple for each k , and $f_j(z_k) \neq 0$ for each j, k . Then the sequence $\{z_{k1}\}$ is convergent.*

The boundedness of $\Lambda_\epsilon(F) \cap \mathbb{C}_{\geq \delta}$ for all $\delta \in \mathbb{R}$ rather than the boundedness of $\Lambda_\epsilon(F)$ is also sufficient for the convergence of the sequence $\{z_k\}$ to a desired point, provided $\|\nabla \lambda_k\|_2$ remains bounded away from zero.

Theorem 2.5 (Convergence). *Suppose that $\Lambda_\epsilon(F) \cap \mathbb{C}_{\geq \delta}$ is bounded for all $\delta \in \mathbb{R}$, $\sigma_{\min}[F(z_k)]$ is simple for each k , and $f_j(z_k) \neq 0$ for each j, k .*

- (i) *If $\nabla \lambda_k \neq 0$ for each k sufficiently large, then $\lambda_k \rightarrow 0$ as $k \rightarrow \infty$.*
- (ii) *If there exists a real scalar $L > 0$ such that $\|\nabla \lambda_k\|_2 > L$ for each k sufficiently large, then*

$$\frac{(1, 0) \cdot \nabla \lambda_k}{\|\nabla \lambda_k\|_2} \rightarrow 1 \quad \text{as } k \rightarrow \infty.$$

The proofs of parts (i) and (ii) are similar to the proofs of Lemma 3.5 and Theorem 3.6 in [18]. Part (i) means that $\mathcal{C}(z_k)$ approaches the boundary of $\Lambda_\epsilon(F)$ as $k \rightarrow \infty$. Moreover, part (ii) amounts to $\nabla \lambda(z_k)$ pointing in the direction of $(1, 0)$ as $k \rightarrow \infty$. Thus, eventually $\mathcal{C}(z_k)$ becomes aligned with the points on the boundary of $\Lambda_\epsilon(F)$ with vertical tangent line. The

assertions of Theorem 2.5 amounts to the satisfaction of the first order optimality conditions by the sequence $\{z_k\}$ in the smooth and nonsmooth sense (i.e., regardless of the multiplicity of $\sigma_{\min}[F(z_*)]$ whenever the limit $z_* = \lim_{k \rightarrow \infty} z_k$ exists).

We conclude this section with a description of the algorithm below. This description is given in the more general rectangular setting, when $F : \Omega \rightarrow \mathbb{C}^{n \times m}$ is analytic on Ω . The ϵ -pseudospectrum can be defined for a rectangular analytic matrix-valued function in a similar fashion by (1.9). The algorithm extends without any modification to this rectangular setting. Throughout this text, in the descriptions of the algorithms we state the termination criteria in exact terms to keep the descriptions neat. Obviously, numerical implementations would require the satisfaction of these conditions up to specified tolerances.

Algorithm 1 Local Search

Require: A matrix-valued function $F : \Omega \rightarrow \mathbb{C}^{n \times m}$ analytic on Ω and a positive scalar $\epsilon \in \mathbb{R}$

- 1: $z_0 \leftarrow (\Re z_R, \Im z_R)$, where z_R is any point in $\Lambda_\epsilon(F)$ and $k \leftarrow 0$.
 - 2: Calculate $\sigma_0 := \sigma_{\min}[F(z_0)]$ and an associated unit right singular vector v_0 .
 - 3: Calculate $\lambda_0, \nabla \lambda_0$ using σ_0, v_0, z_0 .
 - 4: **while** $(\lambda_k \neq 0)$ or $(\nabla \lambda_k \neq c \cdot (1, 0) \ \forall c \in \mathbb{R}^+)$ **do**
 - 5: Apply the recurrence (2.5) to find z_{k+1} given $z_k, \lambda_k, \nabla \lambda_k$.
 - 6: Calculate $\sigma_{k+1} := \sigma_{\min}[F(z_{k+1})]$ and an associated unit right singular vector v_{k+1} .
 - 7: Calculate $\lambda_{k+1}, \nabla \lambda_{k+1}$ using $\sigma_{k+1}, v_{k+1}, z_{k+1}$.
 - 8: Increment k .
 - 9: **end while**
 - 10: **Output:** z_k .
-

3 Vertical Search

It is essential that $\mathcal{C}(z_0) \in \Lambda_\epsilon(F)$ for the locally convergent algorithm of the previous section. In this section, we further impose $\mathcal{C}(z_0)$ to be the rightmost eigenvalue of $F(\lambda)$. This turns out to be essential for global convergence. The sequence $\{z_k\}$ defined by the update rule (2.5), when it converges, yields a point $z_* = (\alpha_*, \beta_*)$ such that

- (1) $\mathcal{C}(z_*)$ is on the boundary of $\Lambda_\epsilon(F)$ with vertical tangent line, or
- (2) $0 \in \partial \lambda(z_*)$.

Above, $\partial \lambda(z_*)$ denotes the generalized gradient of λ at z_* given by [3, page 11]

$$\partial \lambda(z_*) := \text{co} \left\{ \lim_{k \rightarrow \infty} \nabla \lambda(\tilde{z}_k) \mid \tilde{z}_k \rightarrow z_*, \tilde{z}_k \notin \Omega \ \forall k \right\},$$

where $\text{co}(H)$ is the convex hull of the set H and Ω is the subset of \mathbb{R}^2 of measure zero on which λ is not differentiable. Case (2) can occur after finitely many iterations if it happens that $\lambda_k = 0$ and $\nabla \lambda_k = 0$ for some k . In this case, $z_\ell = z_k$ for each $\ell > k$ due to the fact that z_{k+1} is the local maximizer of (2.4). In the more probable infinite convergence case, unless $0 \in \partial \lambda(z_*)$, the point $\mathcal{C}(z_*)$ must be on the boundary of $\Lambda_\epsilon(F)$ with a vertical tangent line by Theorem 2.5.

The point $\mathcal{C}(z_*)$ may or may not be a rightmost point globally in $\Lambda_\epsilon(F)$. To check whether $\mathcal{C}(z_*)$ is indeed a rightmost point globally in $\Lambda_\epsilon(F)$, we globally minimize

$$\sigma(\alpha_*, \omega) := \frac{\sigma_{\min}[F(\alpha_*, \omega)]}{\|w(\alpha_*, \omega)\|_1}$$

over all $\omega \in \mathbb{R}$. This global minimization is achieved by means of the algorithm in [19] for the optimization of a prescribed eigenvalue of a Hermitian and analytic matrix-valued function. If the globally smallest value of $\sigma(\alpha_*, \omega)$ is ϵ , then $\mathcal{C}(z_*)$ is indeed a rightmost point of $\Lambda_\epsilon(F)$ globally. We draw this conclusion based on the assumption that z_0 is the rightmost eigenvalue

of F , and by the fact that each connected component of $\Lambda_\epsilon(F)$ must contain an eigenvalue. If the globally minimal value of $\sigma(\alpha_*, \omega)$ is strictly less than ϵ , then we repeat the locally convergent algorithm of the previous section starting from (α_*, ω_*) , where ω_* is the computed global minimizer of $\sigma(\alpha_*, \omega)$. The point $\mathcal{C}(\alpha_*, \omega_*)$ lies strictly inside $\Lambda_\epsilon(F)$.

The algorithm in [19] to minimize $\sigma(\alpha_*, \omega)$ over ω requires

$$\frac{\partial \sigma(\alpha_*, \omega)}{\partial \omega} = \frac{1}{\|w(\alpha_*, \omega)\|_1} \cdot \Re \left(u^* \frac{\partial F(\alpha_*, \omega)}{\partial \omega} v \right) - \frac{1}{\|w(\alpha_*, \omega)\|_1^2} \cdot \frac{\partial \|w(\alpha_*, \omega)\|_1}{\partial \omega} \cdot \sigma_{\min}[F(\alpha_*, \omega)]$$

where u, v represent a consistent pair of unit left and unit right singular vectors associated with $\sigma_{\min}[F(\alpha_*, \omega)]$, whenever $\sigma_{\min}[F(\alpha_*, \omega)]$ is simple and $f_j(\alpha_*, \omega) \neq 0$ for each $j \in \{0, \dots, m\}$. A lower bound on the second derivatives of $\sigma(\alpha_*, \omega)$, that is a scalar γ satisfying $\partial^2 \sigma(\alpha_*, \omega) / \partial \omega^2 \geq \gamma$ for all ω such that $\sigma(\alpha_*, \omega)$ is differentiable, is also necessary. Unlike the previous section which offered analytical means to choose γ_λ and γ_w , analytical determination of such γ does not seem easy. The additional difficulty is due to a lower bound sought, rather than an upper bound, on the second derivatives of a smallest eigenvalue function. In practice, assigning a large negative real value to γ works robustly.

A description of the vertical search combined with the local search is given in Algorithm 2 below. This yields a globally convergent algorithm to compute $\alpha_\epsilon(F)$. Vertical searches also apply regardless of whether F is a square or a rectangular matrix-valued function.

Algorithm 2 Computation of ϵ -pseudospectral Abscissa for Matrix-Valued Functions

Require: A matrix-valued function $F : \Omega \rightarrow \mathbb{C}^{n \times m}$ analytic on Ω and a positive scalar $\epsilon \in \mathbb{R}$

- 1: $z_0 \leftarrow (\Re z_R, \Im z_R)$, where z_R is a rightmost eigenvalue of F .
 - 2: $Convergence \leftarrow \text{False}$.
 - 3: **while** $\neg Convergence$ **do**
 - 4: **Local Search:** Apply Algorithm 1 starting from z_0 to find $z_* = (\alpha_*, \beta_*)$ such that $\mathcal{C}(z_*) \in \partial \Lambda_\epsilon(F)$ with a vertical tangent line (or $0 \in \partial \lambda(z_*)$).
 - 5: **Vertical Search:** $\omega_* \leftarrow \arg \min_{\omega \in \mathbb{R}} \sigma(\alpha_*, \omega)$ and $\sigma_* \leftarrow \sigma(\alpha_*, \omega_*)$.
 - 6: **if** $\sigma_* = \epsilon$ **then**
 - 7: $Convergence \leftarrow \text{True}$.
 - 8: **else**
 - 9: $z_0 \leftarrow (\alpha_*, \omega_*)$.
 - 10: **end if**
 - 11: **end while**
 - 12: **Output:** z_* .
-

4 Subspace Methods

To cope with large scale problems, we consider the map $v \mapsto F(\lambda)v$ when its domain is restricted to a subspace \mathcal{S} of \mathbb{C}^n . Let S be an isometry (i.e., S is a matrix with more rows than columns satisfying $S^*S = I$) whose columns form an orthonormal basis for \mathcal{S} . The matrix representation of the linear map acting on \mathcal{S} with respect to this basis becomes

$$F_S(\lambda) := F(\lambda)S = \sum_{j=0}^m f_j(\lambda)A_jS.$$

Such a subspace idea is introduced in [11] for the computation of the ϵ -pseudospectral abscissa of a matrix, i.e., when $F(\lambda) = \lambda I - A$. Here we extend it to the general setting when $F(\lambda)$ is an analytic matrix-valued function of the form (1.2). In this section, we use the following

definitions of the ϵ -pseudospectrum and the ϵ -pseudospectral abscissa of F_S :

$$\Lambda_\epsilon(F_S) := \left\{ \lambda \in \mathbb{C} : \sigma_{\min} \left(\sum_{j=0}^m f_j(\lambda) A_j S \right) \leq \epsilon \|w(\lambda)\|_1 \right\} \quad \text{and}$$

$$\alpha_\epsilon(F_S) := \sup \left\{ \Re \lambda : \lambda \in \Lambda_\epsilon(F_S) \right\}.$$

These terminologies and notations are slightly illusive. Indeed, the set $\Lambda_\epsilon(F_S)$ and the quantity $\alpha_\epsilon(F_S)$ are intrinsic to the underlying linear map acting on \mathcal{S} , and independent of the choice of the orthonormal basis (given by the columns of S) for \mathcal{S} . We pursue them in order to remain consistent with the previous sections.

The use of restrictions to subspaces is justified by Corollary 4.3 below, which shows the existence of a unit vector v such that $\alpha_\epsilon(F) = \alpha_\epsilon(F_v)$.

Lemma 4.1 (Monotonicity). *Two isometries S_1, S_2 such that $\text{Col}(S_1) \subseteq \text{Col}(S_2)$ satisfy*

$$(1) \quad \Lambda_\epsilon(F_{S_1}) \subseteq \Lambda_\epsilon(F_{S_2}) \quad \text{and} \quad (2) \quad \alpha_\epsilon(F_{S_1}) \leq \alpha_\epsilon(F_{S_2}).$$

Proof. Let $\mathcal{S}_j := \text{Col}(S_j)$ for $j = 1, 2$. Suppose $z \in \Lambda_\epsilon(F_{S_1})$, that is

$$\sigma_{\min} \left(\sum_{j=0}^m f_j(z) A_j S_1 \right) \leq \epsilon \|w(z)\|_1 \tag{4.1}$$

holds. Notice that

$$\begin{aligned} \sigma_{\min} \left(\sum_{j=0}^m f_j(z) A_j S_1 \right) &= \min_{v \in \mathcal{S}_1, \|v\|_2=1} \left\| \sum_{j=0}^m f_j(z) A_j v \right\|_2 \geq \\ &\min_{v \in \mathcal{S}_2, \|v\|_2=1} \left\| \sum_{j=0}^m f_j(z) A_j v \right\|_2 = \sigma_{\min} \left(\sum_{j=0}^m f_j(z) A_j S_2 \right), \end{aligned}$$

where the inequality is due to $\mathcal{S}_1 \subseteq \mathcal{S}_2$. Combining this with inequality (4.1), we deduce that $z \in \Lambda_\epsilon(F_{S_2})$ proving (1). Furthermore, (2) is an immediate consequence of (1). \square

Theorem 4.2. *Let $z \in \Lambda_\epsilon(F)$ and v be a unit right singular vector associated with $\sigma_{\min}[F(z)]$. Then $z \in \Lambda_\epsilon(F_v)$.*

Proof. The point z satisfies

$$\sigma_{\min} \left(\sum_{j=0}^m f_j(z) A_j \right) \leq \epsilon \|w(z)\|_1. \tag{4.2}$$

Furthermore,

$$\sigma_{\min} \left(\sum_{j=0}^m f_j(z) A_j \right) = \left\| \sum_{j=0}^m f_j(z) A_j v \right\|_2 = \sigma_{\min} \left(\sum_{j=0}^m f_j(z) A_j v \right).$$

Thus, by the inequality in (4.2), we have $z \in \Lambda_\epsilon(F_v)$ as desired. \square

Corollary 4.3 (Low Dimensionality). *Let z_* be a globally rightmost point of $\Lambda_\epsilon(F)$ and v_* be a unit right singular vector associated with $\sigma_{\min}[F(z_*)]$. We have $\alpha_\epsilon(F) = \alpha_\epsilon(F_{v_*})$.*

Proof. By Theorem 4.2, we have $z_* \in \Lambda_\epsilon(F_{v_*})$. This implies that

$$\alpha_\epsilon(F) = \Re z_* \leq \alpha_\epsilon(F_{v_*}).$$

Assertion (2) of Lemma 4.1 implies the reverse of this inequality, yielding $\alpha_\epsilon(F) = \alpha_\epsilon(F_{v_*})$ as desired. \square

The next result shows that for any properly chosen subspace \mathcal{S} containing the optimal right singular vector and any isometry S such that $\text{Col}(S) = \mathcal{S}$, the equality $\alpha_\epsilon(F) = \alpha_\epsilon(F_S)$ holds. Below, $\mathcal{V}(z)$ denotes the set consisting of right singular vectors corresponding to $\sigma_{\min}[F(z)]$ for a given $z \in \mathbb{C}$.

Corollary 4.4. *Suppose that S is an isometry. Then $\alpha_\epsilon(F) = \alpha_\epsilon(F_S)$ if and only if $\mathcal{V}(z_*) \cap \text{Col}(S) \neq \emptyset$ for some globally rightmost point z_* of $\Lambda_\epsilon(F)$.*

Proof. Suppose $\mathcal{V}(z_*) \cap \text{Col}(S) \neq \emptyset$ for some globally rightmost point z_* of $\Lambda_\epsilon(F)$. Consider any $v_* \in \mathcal{V}(z_*) \cap \text{Col}(S)$. Without loss of generality, we can assume v_* is a unit vector. We have $\text{span}\{v_*\} \subseteq \text{Col}(S)$, so by assertion (2) of Lemma 4.1, we obtain

$$\alpha_\epsilon(F_{v_*}) \leq \alpha_\epsilon(F_S) \leq \alpha_\epsilon(F).$$

But from Corollary 4.3, we also have $\alpha_\epsilon(F_{v_*}) = \alpha_\epsilon(F)$ leading to $\alpha_\epsilon(F_S) = \alpha_\epsilon(F)$ as desired.

To prove the converse, suppose $\alpha_\epsilon(F) = \alpha_\epsilon(F_S)$. Denote a globally rightmost point of $\Lambda_\epsilon(F_S)$ with z_S . Due to assertion (1) of Lemma 4.1, we have $z_S \in \Lambda_\epsilon(F)$ and $\Re z_S = \alpha_\epsilon(F_S) = \alpha_\epsilon(F)$, so z_S is also a rightmost point of $\Lambda_\epsilon(F)$ globally. Furthermore, letting v_S be a unit right singular vector associated with $\sigma_{\min}[F_S(z_S)]$, observe

$$\left\| \sum_{j=0}^m f_j(z_S) A_j S v_S \right\|_2 = \sigma_{\min}[F_S(z_S)] = \sigma_{\min}[F(z_S)].$$

Thus $S v_S$ is a unit right singular vector associated with $\sigma_{\min}[F(z_S)]$, that is $\mathcal{V}(z_S) \cap \text{Col}(S) \neq \emptyset$, completing the proof. \square

Corollaries 4.3 and 4.4 give the initiative to work on an $n \times p$ matrix-valued function F_S for $p \ll n$ and for a properly chosen subspace $\mathcal{S} = \text{Col}(S)$ (rather than working on the full $n \times n$ matrix-valued function F). A natural choice for \mathcal{S} appears to be the span of right singular vectors of $\sigma_{\min}[F(z)]$ for various $z \in \mathbb{C}$ close to globally rightmost points of $\Lambda_\epsilon(F)$. The following observations lead us to this choice: (i) *the right singular vectors of $\sigma_{\min}[F(z)]$ are continuous w.r.t. z - if $z \approx z_*$, then $\mathcal{V}(z) \approx \mathcal{V}(z_*)$* ; (ii) *$\alpha_\epsilon(F_S)$ is continuous w.r.t. $\text{Col}(S)$ - if $\text{Col}(S) \approx \text{Col}(S_*)$ for any isometry S_* such that $\text{Col}(S_*) \cap \mathcal{V}(z_*) \neq \emptyset$, then $\alpha_\epsilon(F_S) \approx \alpha_\epsilon(F_{S_*}) = \alpha_\epsilon(F)$* . Initially, we could consider the rightmost eigenvalue z_R of F as a good approximation for the globally rightmost point of $\Lambda_\epsilon(F)$ (this is especially true for $\epsilon \approx 0$). Setting $z_0 := z_R$, we could then generate a sequence $\{z_k\}$ in \mathbb{C} such that

Framework 1 (Subspace Selection Based on Smallest Singular Value)

- (1) S_k is an isometry s.t. its columns form an orthonormal basis for $\mathcal{S}_k := \text{span}\{v_0, \dots, v_k\}$, where v_j is a right singular vector corresponding to $\sigma_{\min}[F(z_j)]$,
 - (2) z_{k+1} is a rightmost point of $\Lambda_\epsilon(F_{S_k})$
-

for $k \in \mathbb{N}$.

Remark. *There is a natural alternative to this way of choosing subspaces. Corollaries 4.3 and 4.4 could be interpreted in terms of eigenvectors corresponding to rightmost eigenvalues of perturbed matrix-valued functions $F + \Delta$. In [21, Proposition 3.1], it is shown that each $z \in \mathbb{C}$ on the boundary of $\Lambda_\epsilon(F)$ is an eigenvalue of*

$$(F + \Delta_z)(\lambda) := \sum_{j=0}^m f_j(\lambda) (A_j + \delta A_{z,j}) \quad \text{where } \delta A_{z,j} := -\frac{\epsilon \cdot \overline{f_j(z)}}{w_j \cdot |f_j(z)|} u v^*,$$

and u, v are compatible unit left, unit right singular vectors corresponding to $\sigma_{\min}[F(z)]$. Indeed, u, v are left, right eigenvectors associated with the eigenvalue z of $F + \Delta_z$. Furthermore, if z_* is a globally rightmost point of $\Lambda_\epsilon(F)$, then it is a rightmost eigenvalue of $F + \Delta_{z_*}$. Thus,

v_* in Corollary 4.3 could also be interpreted as an eigenvector of a rightmost eigenvalue of $F + \Delta_{z_*}$. Similarly, $\mathcal{V}(z_*)$ in Corollary 4.4 could be interpreted as the set of all eigenvectors associated with a rightmost eigenvalue of $F + \Delta_{z_*}$. These observations suggest forming the subspace \mathcal{S} from eigenvectors associated with a rightmost eigenvalue of $F + \Delta_z$ for various z close to globally rightmost points of $\Lambda_\epsilon(F)$. Setting $z_0 := z_R$ the rightmost eigenvalue of F , an alternative sequence $\{z_k\}$ in \mathbb{C} is defined by:

Alternative Framework for Subspace Selection

- (1) S_k is an isometry s.t. its columns form an orthonormal basis for $\mathcal{S}_k := \text{span}\{v_0, \dots, v_k\}$, where v_j is an eigenvector associated with a rightmost eigenvalue of $F + \Delta_{z_j}$,
 - (2) z_{k+1} is a rightmost point of $\Lambda_\epsilon(F_{S_k})$
-

for $k \in \mathbb{N}$. Unfortunately, numerical estimation of a rightmost eigenvalue is harder than that of a smallest singular value. More importantly, the alternative framework above requires the solution of a nonlinear eigenvalue problem, unlike Framework 1 which is based on a standard singular value problem. Retrieving rightmost eigenvalues of nonlinear eigenvalue problems can be computationally expensive in some cases. Thus, we abandon this alternative framework, rather we adopt Framework 1 based on standard smallest singular value computations.

Below, we illustrate how the subspace idea can be put in use for large-scale F by coupling Framework 1 above with Algorithm 1 of Section 2. This results in Algorithm 3 that returns a locally rightmost point of $\Lambda_\epsilon(F)$. In the description of this algorithm, we adopt the notation $\sigma(\Re z, \Im z) := \sigma_{\min}[F(\Re z, \Im z)] / \|w(\Re z, \Im z)\|_1$. We disregard the possibility that the local search (on line 5) converges to a point where the generalized gradient of λ (associated with F_{S_k}) contains zero, which is extremely unlikely. Such an unlikely case can be dealt with for instance by occasional vertical searches at additional cost. Furthermore, when $\sigma(z_{k+1}) = \epsilon$ holds, the condition $c \cdot (1, 0) \in \partial\sigma(z_{k+1}) \ \exists c \in \mathbb{R}^+$ on line 6 (recall that $\partial\sigma(z_{k+1})$ represents the generalized gradient of σ at z_{k+1}) amounts to having a vertical tangent line on the boundary of $\Lambda_\epsilon(F)$ at z_{k+1} in the nonsmooth sense. In the smooth case, when $\sigma(z_{k+1})$ is simple, this condition reduces to $\nabla\sigma(z_{k+1}) = c \cdot (1, 0) \ \exists c \in \mathbb{R}^+$.

Algorithm 3 Large-Scale Local Search

Require: A matrix-valued function $F : \Omega \rightarrow \mathbb{C}^{n \times n}$ analytic on Ω and a positive scalar $\epsilon \in \mathbb{R}$

- 1: $z_0 \leftarrow$ a rightmost eigenvalue of F and $k \leftarrow 0$.
- 2: $\mathcal{S}_0 \leftarrow \text{span}\{v_0\}$, where v_0 is a right singular vector associated with $\sigma_{\min}[F(z_0)]$.
- 3: *Convergence* \leftarrow False.
- 4: **while** \neg *Convergence* **do**
- 5: **Local Search:** Apply Algorithm 1 to find $z_{k+1} = (\alpha_*, \beta_*)$ such that $\mathcal{C}(z_{k+1}) \in \partial\Lambda_\epsilon(F_{S_k})$ with a vertical tangent line.
- 6: **if** $(\sigma(z_{k+1}) = \epsilon)$ and $(c \cdot (1, 0) \in \partial\sigma(z_{k+1}) \ \exists c \in \mathbb{R}^+)$ **then**
- 7: *Convergence* \leftarrow True.
- 8: **else**
- 9: $\mathcal{S}_{k+1} \leftarrow \text{span}(\mathcal{S}_k \cup \{v_{k+1}\})$, where v_{k+1} is a right singular vector associated with $\sigma_{\min}[F(z_{k+1})]$.
- 10: Increment k .
- 11: **end if**
- 12: **end while**
- 13: **Output:** z_{k+1} .

5 Restarts

In the subspace $\mathcal{S}_k = \text{span}\{v_0, \dots, v_k\}$ of Framework 1 the vectors added lately are more relevant to the set of optimal right singular vectors $\mathcal{V}(z_*)$ of Corollary 4.4. This is because the sequence

$\{\Re z_k\}$ is monotone increasing and later points in $\{z_k\}$ usually represent the optimal z_* better. This brings up a subspace-restart idea: when the subspace \mathcal{S}_k becomes of high dimension, erase off the earlier vectors and keep the lastly added few vectors among v_0, \dots, v_k , possibly only v_k . Thus redefine $\mathcal{S}_j := \text{span}\{v_{k-j}, \dots, v_k\}$ and restart. Such restart strategies have already been employed in the context of large scale eigenvalue computation based on Krylov subspace methods [15], and incorporated into modern software, for instance ARPACK [16].

The vertical searches described in Section 3 can also benefit from this restart strategy. A vertical search, when it determines that a point is not globally rightmost, it also provides a new point z_0 to start with. Then the subspace could be reset to $\text{span}\{v_0\}$ where v_0 is a right singular vector associated with $\sigma_{\min}[F(z_0)]$.

5.1 Quality of Subspace Approximations

To further motivate the restart strategy and discarding poor approximations in the subspace, below we relate the quality of the subspace \mathcal{S}_k (specifically its proximity to v_*) with the quality of $\alpha_\epsilon(F_{\mathcal{S}_k})$ (specifically its proximity to $\alpha_\epsilon(F)$). Here and throughout this subsection \mathcal{S}_k , S_k are as in Framework 1, and z_* , v_* are as defined in Corollary 4.3. Furthermore, let u_* be a unit left singular vector associated with $\sigma_{\min}[F(z_*)]$, and consistent with v_* , that is

$$F(z_*)v_* = \sigma_{\min}[F(z_*)]u_* \quad \text{and} \quad u_*^*F(z_*) = \sigma_{\min}[F(z_*)]v_*^*.$$

We measure the quality of \mathcal{S}_k in terms of $\delta = v_* - v_{k*}$ where $v_{k*} := \arg \min_{v \in \mathcal{S}_k, \|v\|_2=1} \|v_* - v\|_2$.

Our approach has two stages. In the first stage, we establish

$$\|F(z_*)v_{k*}\|_2 / \|w(z_*)\|_1 = \epsilon + O(\|\delta\|_2^2). \quad (5.1)$$

In the second stage, starting from this equality, we deduce the existence of a $z_{k*} \in \mathbb{C}$ satisfying

$$\|F(z_{k*})v_{k*}\|_2 / \|w(z_{k*})\|_1 = \epsilon \quad (5.2)$$

and $\Re z_* - \Re z_{k*} = O(\|\delta\|_2^2)$. The last equality implies that $z_{k*} \in \Lambda_\epsilon(F_{\mathcal{S}_k})$, meaning $\Re z_{k*} \leq \alpha_\epsilon(F_{\mathcal{S}_k})$. The desired relation between the ϵ -pseudospectral abscissa of F and $F_{\mathcal{S}_k}$ follows from $\alpha_\epsilon(F) - \alpha_\epsilon(F_{\mathcal{S}_k}) \leq \Re z_* - \Re z_{k*}$.

To prove equality (5.1), let us define the vector-valued function

$$v : \mathbb{R} \rightarrow \mathbb{R}^n, \quad v(t) := \left[v_* + \frac{v_{k*} - v_*}{\|v_{k*} - v_*\|_2} t \right] / \left\| v_* + \frac{v_{k*} - v_*}{\|v_{k*} - v_*\|_2} t \right\|_2,$$

and the scalar function

$$\mu : \mathbb{R} \rightarrow \mathbb{R}, \quad \mu(t) := \|F(z_*)v(t)\|_2,$$

which is real analytic near 0. We benefit from a Taylor expansion of $\mu(t)$ about 0 to obtain (5.1). Specifically, since $v(\|\delta\|_2) = v(\|v_{k*} - v_*\|_2) = v_{k*}$, we have

$$\begin{aligned} \frac{\|F(z_*)v_{k*}\|_2}{\|w(z_*)\|_1} &= \frac{\|F(z_*)v(\|\delta\|_2)\|_2}{\|w(z_*)\|_1} = \frac{\mu(\|\delta\|_2)}{\|w(z_*)\|_1} \\ &= \frac{\mu(0) + \mu'(0)\|\delta\|_2 + O(\|\delta\|_2^2)}{\|w(z_*)\|_1}. \end{aligned}$$

The desired equality (5.1) follows from the observations $\mu(0) = \sigma_{\min}[F(z_*)] = \epsilon\|w(z_*)\|_1$ (since $v(0) = v_*$), and

$$\mu'(0) = \Re(u_*^*F(z_*)v'(0)) = \Re(\sigma_{\min}[F(z_*)]v_*^*v'(0)) = \epsilon\|w(z_*)\|_1\Re(v_*^*v'(0)) = 0,$$

where $\Re(v_*^*v'(0)) = 0$ due to $\|v(t)\|_2^2 = 1$ for all t .

To establish (5.2) for some $z_{k*} \in \mathbb{C}$ such that $\Re z_* - \Re z_{k*} = O(\|\delta\|_2^2)$, let us suppose that $\sigma_{\min}[F(z_*)]$ is simple, and $f_j(z_*) \neq 0$ for $j = 0, \dots, m$. In this case, all of the functions

$$\begin{aligned} \sigma(\Re z, \Im z) &:= \frac{\sigma_{\min}[F(\Re z, \Im z)]}{\|w(\Re z, \Im z)\|_1}, & \hat{\sigma}(\Re z, \Im z) &:= \frac{\|F(\Re z, \Im z)v_*\|_2}{\|w(\Re z, \Im z)\|_1} \\ \text{and } \tilde{\sigma}(\Re z, \Im z) &:= \frac{\|F(\Re z, \Im z)v_{k*}\|_2}{\|w(\Re z, \Im z)\|_1} \end{aligned}$$

are continuously differentiable at $(\Re z, \Im z) = (\Re z_*, \Im z_*)$. Target equality (5.2) can be written as $\tilde{\sigma}(\Re z_{k*}, \Im z_{k*}) = \epsilon$, whereas (5.1) can be expressed as

$$\tilde{\sigma}(\Re z_*, \Im z_*) = \epsilon + O(\|\delta\|_2^2). \quad (5.3)$$

Assuming that the gradient of $\sigma(\Re z, \Im z)$ does not vanish at $(\Re z_*, \Im z_*)$, the first order optimality conditions imply

$$\nabla \sigma(\Re z_*, \Im z_*) = \nabla \hat{\sigma}(\Re z_*, \Im z_*) = c \cdot (1, 0) \quad (5.4)$$

for some positive $c \in \mathbb{R}$. But then, by continuity and the second equality in (5.4), we have

$$\eta := p^T \nabla \tilde{\sigma}(\Re z_*, \Im z_*) < 0, \quad \text{where } p = -(1, 0). \quad (5.5)$$

Employing (5.3) and due to (5.5), there exists $\alpha > 0$ such that

$$\tilde{\sigma}(\Re z_* - \alpha, \Im z_*) = \frac{\|F(\Re z_* - \alpha, \Im z_*)v_{k*}\|_2}{\|w(\Re z_* - \alpha, \Im z_*)\|_1} = \epsilon.$$

Defining z_{k*} by $\Re z_{k*} := \Re z_* - \alpha$ and $\Im z_{k*} := \Im z_*$, we deduce (5.2). To quantify $\alpha = \Re z_* - \Re z_{k*}$, we expand $\tilde{\sigma}(\Re z, \Im z)$ about $(\Re z_*, \Im z_*)$ only varying $\Re z$ (note that $\tilde{\sigma}(\Re z, \Im z_*)$ is real analytic with respect to $\Re z$ near $\Re z_*$). This leads to

$$\tilde{\sigma}(\Re z_* - \alpha, \Im z_*) = \tilde{\sigma}(\Re z_*, \Im z_*) + \eta \alpha + O(\alpha^2) \implies \alpha = O(\|\delta\|_2^2).$$

Finally, since $v_{k*} \in \mathcal{S}_k$, we have

$$\frac{\sigma_{\min}[F_{S_k}(z_{k*})]}{\|w(z_{k*})\|_1} \leq \frac{\|F(z_{k*})v_{k*}\|_2}{\|w(z_{k*})\|_1} = \epsilon$$

meaning $z_{k*} \in \Lambda_\epsilon(F_{S_k})$. From $\alpha_\epsilon(F) = \Re z_*$ and $\alpha_\epsilon(F_{S_k}) \geq \Re z_{k*}$, we obtain $\alpha_\epsilon(F) - \alpha_\epsilon(F_{S_k}) \leq \Re z_* - \Re z_{k*} = \alpha$. Hence,

$$\alpha_\epsilon(F) - \alpha_\epsilon(F_{S_k}) = O(\|\delta\|_2^2).$$

5.2 Overall Algorithm

We apply the subspace method, specifically Framework 1 in Section 4. Initially, z_0 is chosen as the rightmost eigenvalue of F , and \mathcal{S}_0 is the associated one-dimensional subspace. The subspace method requires the determination of a rightmost point of $\Lambda_\epsilon(F_{S_k})$ for several k , each of which we achieve by the local algorithm in Section 2. In practice, this results in convergence to a point $z_* = (z_{*1}, z_{*2}) \in \mathbb{R}^2$ such that $\mathcal{C}(z_*)$ is (up to a tolerance) on the boundary of $\Lambda_\epsilon(F)$ with a vertical tangent line for a rather small subspace \mathcal{S}_k . We do not allow \mathcal{S}_k to expand arbitrarily and restart with a one-dimensional subspace once its dimension reaches a prescribed value. We perform a vertical search discussed in Section 3 along the line $\{z \in \mathbb{C} \mid \Re z = z_{*1}\}$. The vertical search is performed on the full matrix-valued function F in order to ensure global convergence. Termination occurs if this vertical search yields ϵ as the smallest value of the singular value function involved. Otherwise, we restart the subspace method from a global minimizer on the vertical line and with the associated one-dimensional subspace. A detailed description is given in Algorithm 4 below.

Algorithm 4 Computation of ϵ -pseudospectral Abscissa for Large-Scale Matrix-Valued Functions

Require: A matrix-valued function $F : \Omega \rightarrow \mathbb{C}^{n \times n}$ analytic on Ω , a positive scalar $\epsilon \in \mathbb{R}$ and the maximal subspace dimension allowed $k_{\max} \in \mathbb{Z}^+$

```

1:  $z_0 \leftarrow$  a rightmost eigenvalue of  $F$  and  $k \leftarrow 0$ .
2:  $\mathcal{S}_0 \leftarrow \text{span}\{v_0\}$ , where  $v_0$  is a right singular vector associated with  $\sigma_{\min}[F(z_0)]$ .
3:  $\text{Convergence} \leftarrow \text{False}$ .
4: while  $\neg \text{Convergence}$  do
5:   Local Search: Apply Algorithm 1 to find  $z_{k+1} = (\alpha_*, \beta_*)$  such that  $\mathcal{C}(z_{k+1}) \in \partial\Lambda_\epsilon(F_{S_k})$ 
     with a vertical tangent line.
6:   if  $(\sigma(z_{k+1}) = \epsilon)$  and  $(c \cdot (1, 0) \in \partial\sigma(z_{k+1}) \ \exists c \in \mathbb{R}^+)$  then
7:     Vertical Search:  $\omega_* \leftarrow \arg \min_{\omega \in \mathbb{R}} \sigma(\alpha_*, \omega)$  and  $\sigma_* \leftarrow \sigma(\alpha_*, \omega_*)$ .
8:     if  $\sigma_* = \epsilon$  then
9:        $\text{Convergence} \leftarrow \text{True}$ .
10:    else
11:       $z_0 \leftarrow \mathcal{C}(\alpha_*, \omega_*)$  and  $k \leftarrow 0$ .
12:       $\mathcal{S}_0 \leftarrow \text{span}\{v_0\}$ , where  $v_0$  is a right singular vector associated with  $\sigma_{\min}[F(z_0)]$ .
13:    end if
14:  else
15:    if  $k = k_{\max}$  then
16:       $z_0 \leftarrow z_{k+1}$  and  $k \leftarrow 0$ .
17:       $\mathcal{S}_0 \leftarrow \text{span}\{v_0\}$ , where  $v_0$  is a right singular vector associated with  $\sigma_{\min}[F(z_0)]$ .
18:    else
19:       $\mathcal{S}_{k+1} \leftarrow \text{span}(\mathcal{S}_k \cup \{v_{k+1}\})$ , where  $v_{k+1}$  is a right singular vector associated with
         $\sigma_{\min}[F(z_{k+1})]$ .
20:      Increment  $k$ .
21:    end if
22:  end if
23: end while
24: Output:  $z_{k+1}$ .

```

6 Numerical Examples

6.1 Standard Eigenvalue Problem

Algorithm 4 is especially suitable for the computation of the pseudospectral abscissa for large-scale matrices, that is when $F(z) = A - zI$ for a given large matrix A and weights are given by $[1 \ \infty]$. The *criss-cross algorithm* [2] is the most reliable choice for the computation of the pseudospectral abscissa of a matrix at the moment, but its use is limited to mainly small up to medium scale matrices. We compare Algorithm 4 with the criss-cross algorithm in this subsection.

In our numerical experiments, Algorithm 4 in practice is terminated after a vertical search, whenever it is determined that the globally smallest value of $\sigma_{\min}[A - (\alpha_* + \omega i)I]$ over ω (for an α_* converged by the local search algorithm) does not differ from ϵ by more than a tolerance, $10^{-6}\|A\|_2$ for the examples below. The criss-cross algorithm also performs vertical searches, but they are based on extracting all imaginary eigenvalues of $2n \times 2n$ Hamiltonian matrices from which the intersection points of a vertical line with the ϵ -pseudospectrum boundary are inferred. It terminates either if a vertical search fails to find any intersection point or if two consecutive estimates for the ϵ -pseudospectral abscissa are not increasing due to rounding errors.

We start with a random 50×50 matrix formed by typing `randn(50) + 0.7*i*randn(50)` in Matlab. Algorithm 4 applies the subspace iteration initially. Each subspace iteration amounts to a local search on a small problem. When the subspace becomes eight dimensional, it stops expanding as the local searches on seven and eight dimensional subspaces return nearly identical rightmost points. Instead, it performs a vertical search and terminates. In Figure 1, the progress of the subspace iteration on this example is shown for two, four and six dimensional subspaces. The results of Algorithm 4 and the criss-cross algorithm match up to 12 decimal digits.

The next three sets of examples illustrate the superiority of Algorithm 4 to the criss-cross algorithm for medium to large scale matrices. All these examples can be generated using EigTool [29]. Each one of the three test sets consists of four matrices of size 200, 400, 800 and 1200 chosen from a particular family. The matrices in the first set are Landau matrices arising from an integral equation in laser theory [14]. The matrices in the second set are Hatano-Nelson matrices, which are tridiagonal and arise from quantum mechanics [8]. The matrices in the third set are Davies matrices originating from a spectral method discretization of an anharmonic oscillator, i.e., a second order differential operator subject to boundary conditions in 1-dimension [4]. A comparison of running times of the algorithms are given in Table 1. In all cases, Algorithm 4 becomes superior in terms of the running times, as soon as n (the size of the matrix) is slightly larger than 200. The gap grows quickly as n increases. Furthermore, roughly the quadratic dependence of the running time for Algorithm 4 on the sizes of the matrices is apparent. This is due to the fact that the required smallest eigenvalues and singular values, and the corresponding eigenvectors and singular vectors are computed by means of ARPACK [16], which is based on Arnoldi's method. The computed results by the two algorithms differ by amounts on the order of the double machine precision in a relative sense. More precisely, denoting the results returned by Algorithm 4 and the criss-cross algorithm with f_1 and f_2 , respectively, and the 2-norm of the input matrix with $\|A\|_2$, the quantity $|f_1 - f_2|/\|A\|_2$ does not exceed 10^{-14} .

6.2 Polynomial Eigenvalue Problem

In this subsection, we experiment with several polynomial eigenvalue problems available in the collection [1]. In all of these experiments, Algorithm 4 is terminated in practice whenever a vertical search determines that the globally smallest value of $\sigma_{\min}[F(\alpha_*, \omega)]/\|w(\alpha_*, \omega)\|_1$ over ω does not differ from ϵ by more than $10^{-6}\|A_m\|_2$, where A_m denotes the leading coefficient matrix of the matrix polynomial $F(\lambda) := \sum_{j=0}^m \lambda^j A_j$ whose ϵ -pseudospectral abscissa is sought. Furthermore, in all experiments in this subsection, all weights are set equal to one, unless otherwise stated.

Wing Example: The first one arises from the analysis of oscillations of a wing of an airplane,

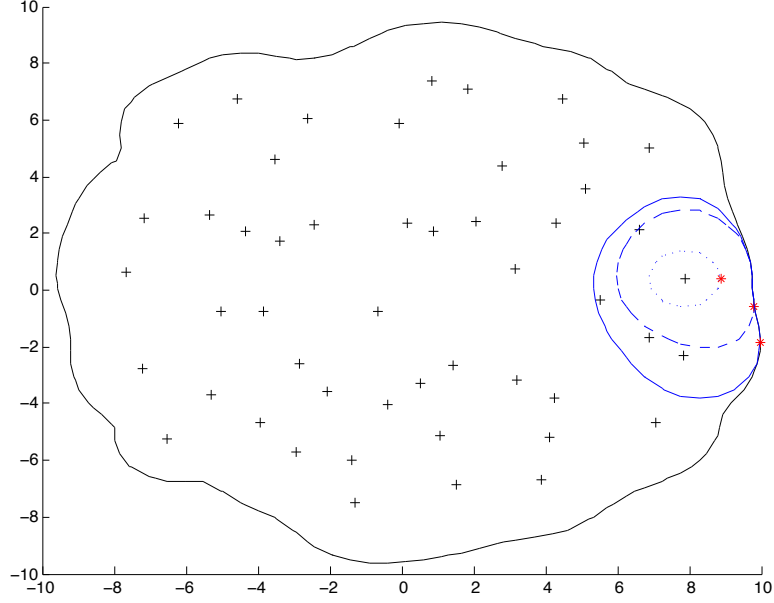


Figure 1: Subspace iteration on a 50×50 random matrix is displayed. The outermost black curve represents the boundary of the ϵ -pseudospectrum of this matrix for $\epsilon = 1$, while each $+$ represents an eigenvalue. The ϵ -pseudospectrum for the restricted problem F_{S_k} is shown with dotted, dashed and solid blue curves for $k = 2, 4, 6$, respectively. The red asterisks mark the rightmost points for these restricted problems.

Landau, $\epsilon = 10^{-0.5}$	200	400	800	1200
Algorithm 4	7	20	78	176
Criss-Cross Algorithm	7	43	223	662
α_ϵ	1.3153	1.3161	1.3161	1.3161

Hatano, $\epsilon = 1$	200	400	800	1200
Algorithm 4	14	29	89	207
Criss-Cross Algorithm	7	46	2030	
α_ϵ	4.0765	4.0903	4.0678	4.1474

Davies, $\epsilon = 10^5$	200	400	800	1200
Algorithm 4	8	9	19	40
Criss-Cross Algorithm	3	41	223	
α_ϵ	$4.0355 \cdot 10^5$	$4.8867 \cdot 10^6$	$7.6266 \cdot 10^7$	$3.8504 \cdot 10^8$

Table 1: Running times for the algorithms in seconds with respect to the sizes of the matrices, and the computed pseudospectral abscissa; The running times of the criss-cross algorithm are omitted for Hatano and Davies matrices of size 1200, since its computations take excessive time.

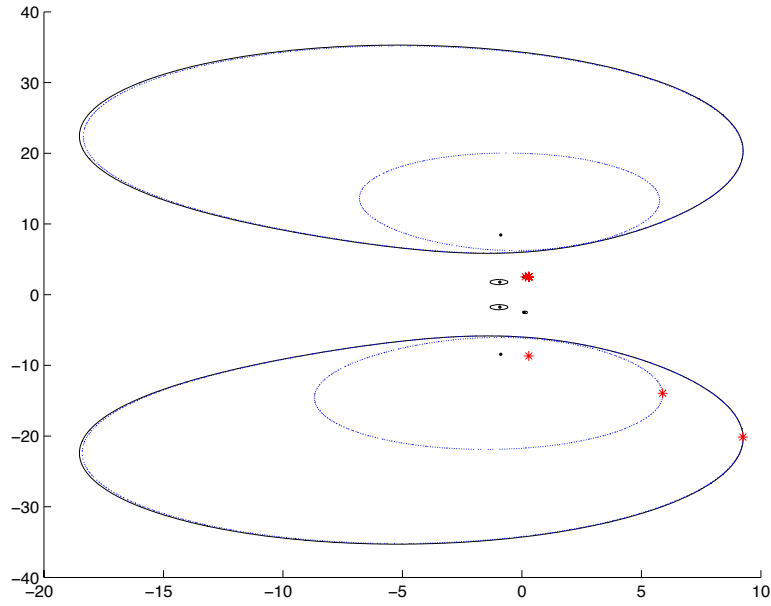


Figure 2: The progress of the algorithm on the wing example Q for $\epsilon = 10^{-0.8}$ is shown. The eigenvalues are marked with black points, and the iterates of the algorithm are marked with red asterisks. The outermost solid curve corresponds to the boundary of $\Lambda_\epsilon(Q)$ for $\epsilon = 10^{-0.8}$, whereas the inner dotted curves represent the boundary of this ϵ -pseudospectrum when the domain of the map $v \mapsto Qv$ is restricted to one and two dimensional subspaces.

and leading to a 3×3 quadratic eigenvalue problem $Q(\lambda) = A_0 + \lambda A_1 + \lambda^2 A_2$ where

$$A_0 = \begin{bmatrix} 121 & 18.9 & 15.9 \\ 0 & 2.7 & 0.145 \\ 11.9 & 3.64 & 15.5 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 7.66 & 2.45 & 2.1 \\ 0.23 & 1.04 & 0.223 \\ 0.60 & 0.756 & 0.658 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 17.6 & 1.28 & 2.89 \\ 1.28 & 0.824 & 0.413 \\ 2.89 & 0.413 & 0.725 \end{bmatrix}.$$

The progress of Algorithm 4 on this example with $\epsilon = 10^{-0.8}$ is illustrated in Figure 2. The algorithm starts with a rightmost eigenvalue $z_r = 0.0947 + 2.2529i$. However, this eigenvalue is considerably less sensitive as compared to the eigenvalues $-0.8848 \pm 8.4415i$. The first few iterations yield estimates in the component of $\Lambda_\epsilon(Q)$ containing z_r . When the rightmost point in this component is obtained, a vertical search is performed, and the algorithm jumps into the component of the eigenvalue $-0.8848 - 8.4415i$. A few more subspace iterations result in the convergence to a rightmost point globally. The computed value $\alpha_\epsilon(Q) = 9.25817665382$ matches the result reported in [21].

Butterfly Example: We next experiment on the butterfly example in [1]. This involves a 64×64 quartic polynomial $P(\lambda) = B_0 + \lambda B_1 + \lambda^2 B_2 + \lambda^3 B_3 + \lambda^4 B_4$ for which the computation of $\alpha_\epsilon(P)$ appears notoriously difficult. A particular application of Algorithm 4 for the computation of $\alpha_\epsilon(P)$ for $\epsilon = 0.08$ is illustrated in Figure 3. The algorithm converges to (nonglobal) local solutions twice. It escapes from these local solutions by means of vertical searches. Consequently, it generates iterates with imaginary parts about -2. But slow convergence occurs, and our numerical implementation applies another vertical search. This is an artifact of the numerical implementation; this vertical search is not essential for convergence to a globally rightmost point, as the pseudospectra is symmetric with respect to the real axis. But it speeds up the convergence. This leads to iterates with imaginary parts about 2, and eventually termination with $\alpha_\epsilon(P) = 1.3858189142$.

To illustrate the effect of the weights, the algorithm is applied to the butterfly example for $\epsilon = 0.2$ with weights $[1 \ 1 \ 1 \ 1]$, $[1 \ 1 \ 1 \ 1 \ \infty]$, $[1 \ 1 \ 1 \ \infty \ \infty]$ and $[1 \ 1 \ \infty \ \infty \ \infty]$. The iterates generated on the associated ϵ -pseudospectrum for each of these four cases are provided in Figure

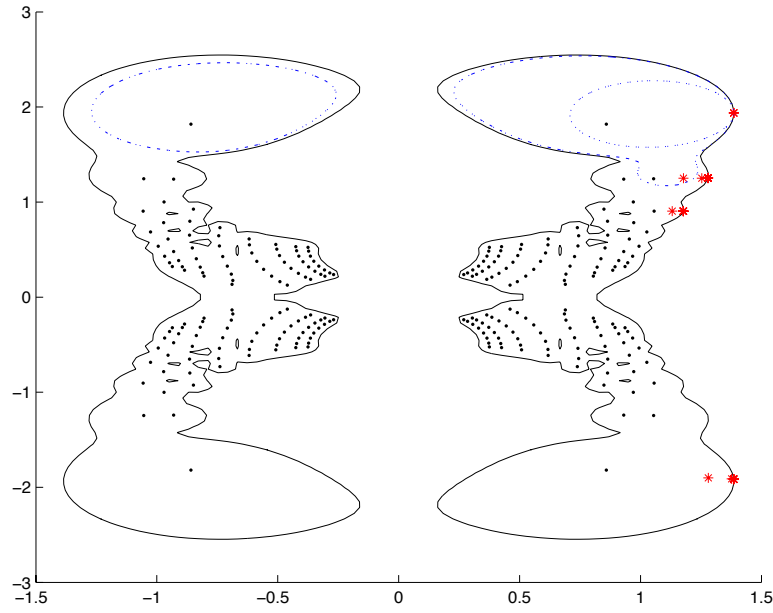


Figure 3: The algorithm is depicted on the butterfly example with $\epsilon = 0.08$. Once again, the eigenvalues and the iterates of the algorithm are marked with black points and red asterisks. The outermost solid curve corresponds to the boundary of ϵ -pseudospectrum. The dotted curve and dashed dotted curve represent the ϵ -pseudospectrum when the domain is restricted to a one dimensional subspace and a three dimensional subspace just a few iterations before termination.

2D Acoustic Wave, $\epsilon = 0.01$	110	210	420
Running Time	11	138	584
α_ϵ	4.99778	6.95044	$1.00718 \cdot 10^1$

Table 2: Running time for Algorithm 4 in seconds on the 2-dimensional acoustic wave equation with respect to the size of the quadratic matrix polynomial involved, and the computed pseudospectral abscissa

4. The computed $\alpha_\epsilon(P)$ are 3.6758307326, 1.4144528011, 1.2006081257 and 1.1221784200, respectively. The decrease in the ϵ -pseudospectral abscissa is dramatic when the perturbations of the leading coefficient are not allowed. The algorithm does not require any vertical searches on the top left figure, it performs vertical searches on the other three figures to avoid locally rightmost points.

2D Acoustic Wave Example: This concerns a quadratic matrix polynomial $W(\lambda) = K_0 + \lambda K_1 + \lambda^2 K_2$ arising from a finite element discretization of a 2-dimensional harmonic wave equation over the unit square $[0, 1] \times [0, 1]$. The size of the matrix polynomial W depends on the coarseness of the finite element grid. Running times of Algorithm 4 to compute $\alpha_\epsilon(W)$ for $\epsilon = 0.01$ with respect to the size of W are listed in Table 2.

6.3 Delay Eigenvalue Problem

We test Algorithm 4 on the following delay eigenvalue problem with weights $[\infty \ 1 \ 1 \ 1]$:

$$D(\lambda) = \lambda I - D_0 - D_1 e^{-\lambda} - D_2 e^{-3\lambda}. \quad (6.1)$$

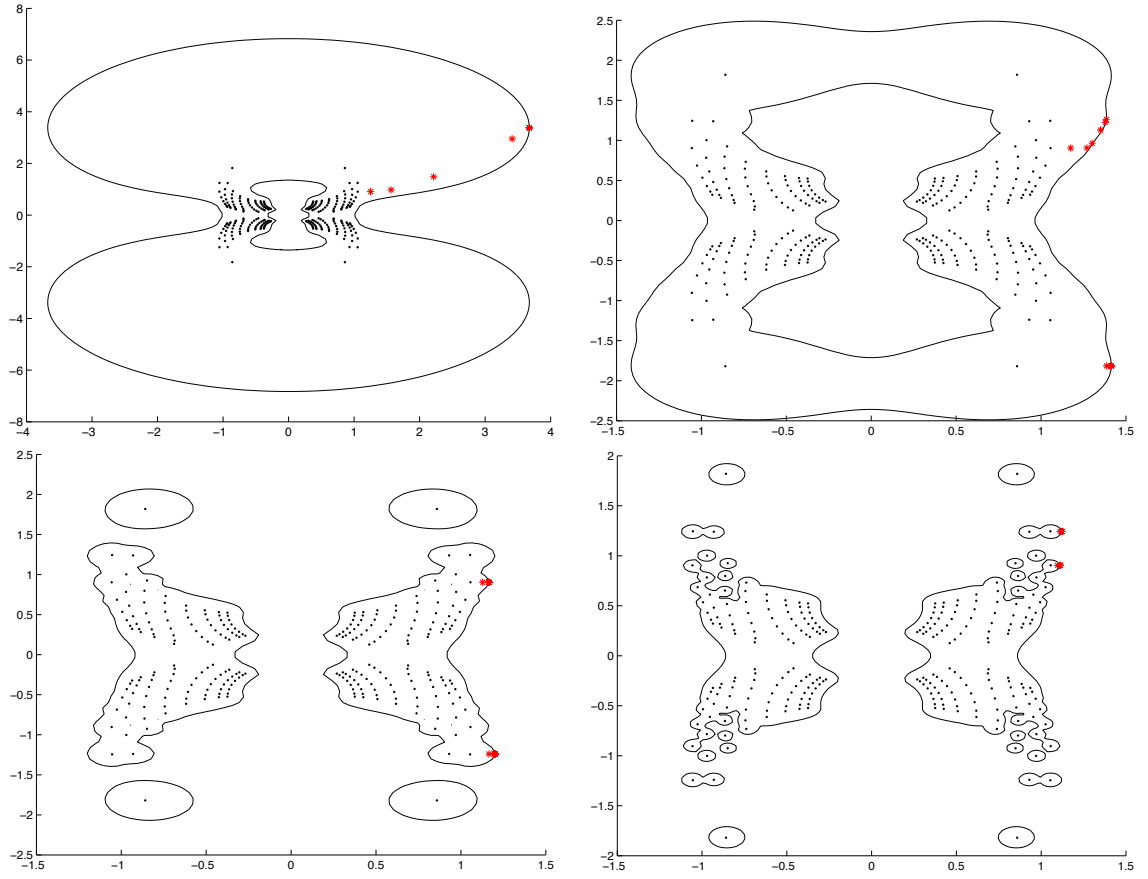


Figure 4: The algorithm on the butterfly example with $\epsilon = 0.2$ is depicted for various choices of weights. The eigenvalues and the iterates are marked with dots and asterisks, respectively. The solid curve is the boundary of the ϵ -pseudospectrum. Weights are as follows: **(top left)** $[1 \ 1 \ 1 \ 1 \ 1]$; **(top right)** $[1 \ 1 \ 1 \ 1 \ \infty]$; **(bottom left)** $[1 \ 1 \ 1 \ \infty \ \infty]$; **(bottom right)** $[1 \ 1 \ \infty \ \infty \ \infty]$.

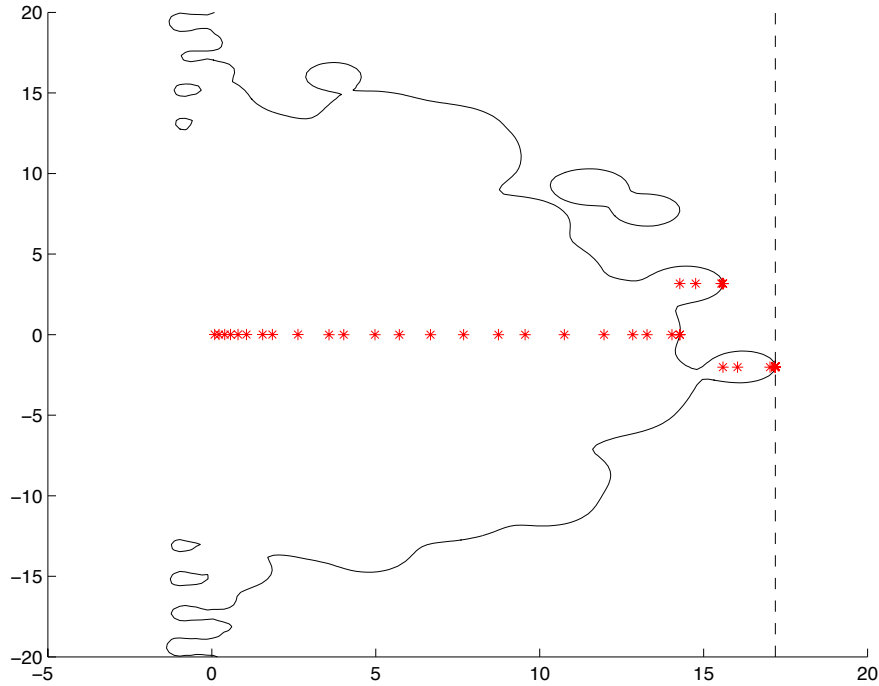


Figure 5: The progress of the algorithm on delay example (6.1) for $\epsilon = 0.45$ is shown. The iterates are marked with red asterisks, whereas the solid curve corresponds to the boundary of the ϵ -pseudospectrum. The dashed vertical line represents the points with real part equal to the computed ϵ -pseudospectral abscissa.

The coefficient matrices D_0, D_1, D_2 are obtained by typing `randn(100) + 1.2*randn(100)*i`, `randn(100)` and `gallery('poisson',10)` in Matlab*. Thus D_0 and D_1 are complex and real random matrices, respectively, whereas D_2 comes from a five-point finite difference discretization of a Poisson equation. The details of the algorithm on this delay example for $\epsilon = 0.45$ are shown in Figure 5. We start the algorithm with the origin as the initial guess. The algorithm ends up at locally rightmost points twice. Each time this happens, a vertical search provides a better estimate strictly inside the ϵ -pseudospectrum and well-away from the boundary. All together 53 subspace iterations are needed to retrieve $\alpha_\epsilon(D) = 17.1899477706$ in about 154 seconds of CPU time. The subspace dimension is never allowed to exceed ten; whenever the subspace dimension becomes ten, it is reset to a one dimensional subspace based on the latest iterate. The algorithm concludes with convergence during the third vertical search, when it is found out that the globally smallest value of $\sigma_{\min}[D(\alpha_*, \omega)] / \|w(\alpha_*, \omega)\|_1$ for fixed $\alpha_* = 17.1899477706$ over all ω does not differ from ϵ by more than $10^{-6} \|D_2\|_2$.

Note that by starting from an arbitrary point λ_0 satisfying $\sigma_{\min}[D(\lambda_0)] < \epsilon \|w(\lambda_0)\|_1$, which we have done to illustrate the role of vertical searches, does by itself not guarantee to find a globally rightmost point of the pseudospectrum, since, for instance, the existence of an isolated component of the pseudospectrum to the right of the dashed line in Figure 5 is not excluded. Such a situation is avoided by initiating the algorithm with the rightmost eigenvalue.

7 Software

Algorithm 4 is implemented in Matlab. This Matlab software is available on the web publicly[†]. For a nonpolynomial nonlinear eigenvalue problem, the user is expected to write down a routine

*The precise data is available on the web at <http://home.ku.edu.tr/~emengi/software/delay.mat>

[†]http://home.ku.edu.tr/~emengi/software/nonlinear_pseudoabscissa.tar

calculating the functions $f_j(z)$ as in (1.2) and its first derivatives at a given $z \in \mathbb{C}$. The user must provide the name of this routine and a right-most eigenvalue to the software as input parameters.

8 Conclusion

An algorithm is proposed for the computation of the ϵ -pseudospectral abscissa of an analytic matrix-valued function $F(\lambda)$ depending on one complex parameter. The algorithm is capable of handling large scale problems. This is made possible by an adaptation of the subspace iteration [11] for the nonlinear eigenvalue problem setting. Each subspace iteration involves the computation of the ϵ -pseudospectral abscissa when the domain of the map $v \mapsto F(\lambda)v$ is restricted to a small subspace. This computation is realized locally, but in a robust way against nonsmoothness, by adapting the support based algorithm of [18] for optimization subject to eigenvalue constraints. Repeated applications of the subspace iteration result in a point on the boundary of the ϵ -pseudospectrum with a vertical tangent line. Vertical searches are performed to check whether these converged points are globally right-most in the ϵ -pseudospectrum. These vertical searches are realized by means of the support based algorithm of [19], which determines the globally smallest value of a prescribed eigenvalue of a Hermitian and analytic matrix-valued function. A novel restarting strategy for the subspaces further enhances the efficiency of the algorithm.

The algorithm is both globally convergent and well-suited for large scale problems. The accompanying software that is publicly available aims for large scale standard, polynomial and more general nonlinear eigenvalue problems.

References

- [1] T. Betcke, N. J. Higham, Mehrmann V., Schroder C., and Tisseur F. NLEVP: A collection of nonlinear eigenvalue problems. Technical Report Eprint 2010.98, Manchester Institute for Mathematical Sciences, 2010.
- [2] J. V. Burke, A. S. Lewis, and M. L. Overton. Robust stability and a criss-cross algorithm for pseudospectra. *IMA J. Numer. Anal.*, 23(3):359–375, 2003.
- [3] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1990.
- [4] E. B. Davies. Pseudo-spectra, the harmonic oscillator and complex resonances. *Proc. Roy. Soc. Lond. Ser. A.*, pages 585–599, 1999.
- [5] C. Effenberg. *Robust Solution Methods for Nonlinear Eigenvalue Problems*. Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, 2013. Ph.D. Thesis.
- [6] K. Green and T. Wagenknecht. Pseudospectra and delay differential equations. *J. Comput. Appl. Math.*, 196(2):567–578, 2006.
- [7] N. Guglielmi and M. L. Overton. Fast algorithms for the approximation of the pseudospectral abscissa and pseudospectral radius of a matrix. *SIAM J. Matrix Anal. Appl.*, 32(4):1166–1192, 2011.
- [8] N. Hatano and D. R. Nelson. Localization transitions in non-Hermitian quantum mechanics. *Phys. Rev. Lett.*, 77:570–573, 1996.
- [9] N. J. Higham and F. Tisseur. More on pseudospectra for polynomial eigenvalue problems and applications in control theory. *Linear Algebra Appl.*, 351-352(1):435 – 453, 2002.
- [10] E. Jarlebring, K. Meerbergen, and W. Michiels. A Krylov method for the delay eigenvalue problem. *SIAM J. Sci. Comput.*, 32(6):3278–3300, 2010.

- [11] D. Kressner and B. Vandereycken. Subspace methods for computing the pseudospectral abscissa and the stability radius. *SIAM J. Matrix Anal. Appl.*, 35(1):292–313, 2014.
- [12] P. Lancaster. On eigenvalues of matrices dependent on a parameter. *Numer. Math.*, 6:377–387, 1964.
- [13] P. Lancaster and P. Psarrakos. On the pseudospectra of matrix polynomials. *SIAM J. Matrix Anal. Appl.*, 27(1):115–129, 2005.
- [14] H. J. Landau. The notion of approximate eigenvalues applied to an integral equation of laser theory. *Quart. Appl. Math.*, 35:165 – 172, 1977.
- [15] R. B. Lehoucq and D. C. Sorensen. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM J. Matrix Anal. Appl.*, 17(4):789–821, 1996.
- [16] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK users’ guide: Solution of large scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, Philadelphia, PA, USA, 1998.
- [17] V. Mehrmann and H. Voss. Nonlinear eigenvalue problems: a challenge for modern eigenvalue methods. *GAMM Mitt. Ges. Angew Math. Mech.*, 27(2):121–152, 2004.
- [18] E. Mengi. A support function based algorithm for optimization with eigenvalue constraints. *SIAM J. Optim.*, 2014. Submitted.
- [19] E. Mengi, E. A. Yildirim, and M. Kilic. Numerical optimization of eigenvalues of Hermitian matrix functions. *SIAM J. Matrix Anal. Appl.*, 35(2):699–724, 2014.
- [20] W. Michiels, K. Green, T. Wagenknecht, and S. I. Niculescu. Pseudospectra and stability radii for analytic matrix functions with application to time-delay systems. *Linear Algebra Appl.*, 418(1):315 – 335, 2006.
- [21] W. Michiels and N. Guglielmi. An iterative method for computing the pseudospectral abscissa for a class of nonlinear eigenvalue problems. *SIAM J. Sci. Comput.*, 34(4):A2366–A2393, 2012.
- [22] W. Michiels and S. I. Niculescu. *Stability and Stabilization of Time-Delay Systems (Advances in Design & Control) (Advances in Design and Control)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007.
- [23] F. Tisseur and N. J. Higham. Structured pseudospectra for polynomial eigenvalue problems, with applications. *SIAM J. Matrix Anal. Appl.*, 23(1):187–208, 2001.
- [24] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Rev.*, 43(2):235–286, 2001.
- [25] L. N. Trefethen and M. Embree. *Spectra and Pseudospectra. The Behavior of Nonnormal Matrices and Operators*. Princeton University Press, Princeton, NJ, USA, 2005.
- [26] R. Van Beeumen. *Rational Krylov Methods for Nonlinear Eigenvalue Problems*. KU Leuven, Leuven, Belgium, 2015. Ph.D. Thesis.
- [27] D. Verhees, R. Van Beeumen, K. Meerbergen, N. Guglielmi, and W. Michiels. Fast algorithms for computing the distance to instability of nonlinear eigenvalue problems, with applications to time-delay systems. *Int. J. Dynam. Control*, 2(4):133 – 142, 2014.
- [28] T. Wagenknecht, W. Michiels, and K. Green. Structured pseudospectra for nonlinear eigenvalue problems. *J. Comput. Appl. Math.*, 212(2):245 – 259, 2008.
- [29] T.G. Wright. Eigtool: a graphical tool for nonsymmetric eigenproblems. Oxford University Computing Laboratory. <https://github.com/eigtool/eigtool>.
- [30] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.